Smile Detection Using Multi-scale Gaussian Derivatives

Varun Jain
INRIA Grenoble Rhône-Alpes and
Université de Grenoble, Grenoble
FRANCE
varun.jain@inria.fr

James L. Crowley
INRIA Grenoble Rhône-Alpes and
Grenoble INP, Grenoble
FRANCE
james.crowley@inria.fr

Abstract: In this paper we show that Multi-scale Gaussian Derivatives combined with Support Vector Machines provide an effective approach to facial expression analysis.
The approach was tested on the GENKI and Cohn-Kanade dataset for detecting smiles. Our approach outperformed the current benchmark approach of using Gabor Energy Filters with Support Vector Machines.

Key–Words: Smile Detection, Multi-scale Gaussian Derivatives, GENKI.

1 Introduction

A lot of research is being done in the field of facial expression recognition. The problem of facial expression recognition has been approached by researchers in two ways: model based approaches and holistic approaches.
In model based approaches facial key points like eyes, eyebrows, nose, lips etc. have to be located and tracked and then the expression is estimated according to the relative position of these facial key points [1, 2, 3, 4].
In holistic or appearance based approaches an image descriptor is used to represent the image and a feature vector is assembled using the descriptor values. Then a suitable machine learning technique is used for discrimination between different facial expressions [5, 6].
A major problem with model based approaches is that key point detection is a difficult task and tracking these key points is all the more likely to fail. With holistic approaches one has to choose an image descriptor and a machine learning technique from a wide array of options. In contrast to the model based approaches one needs training and testing data to make such approaches work but these approaches do not suffer from issues like facial key point detection and tracking failure.
Littlewort in [7] showed that Gabor Energy Filters [8] and Support Vector Machines [9] provide the greatest accuracy for facial action recognition and they showed that by applying these to the popular Cohn-Kanade [10] and POFA [11] datasets.
We in this paper employ Multi-scale Gaussian Derivatives(MGD) for smile detection on the GENKI-4k dataset [12] and show that our choice of descriptor gives better results than Gabor Energy Filters when combined with the Support Vector Machine(SVM).
In the next section we briefly describe the problem of smile detection and the significance of the GENKI-4k dataset.

2 Smile Detection and the GENKI-4k dataset

Smile Detection is the process of inferring whether the person in the image or video is smiling or not. In this paper we will treat it as a binary classification problem.
We need to choose an appropriate descriptor to extract features from the image and then a pattern recognition algorithm is required to discriminate between the 'smiling' and the 'not-smiling' images.
The approach that we present in the following sections was tested on the GENKI-4k dataset [12]. This dataset contains 4,000 face images spanning a wide range of subjects, facial appearence, illumination, geographical locations, imaging conditions and camera models. All images are labeled for smile content(1=smile, 0=non-smile).
The difference between this dataset and other facial expression datasets is that this dataset was compiled from images on the internet and not produced in a controlled environment [13].

ISBN: 978-1-61804-164-7

Figure 1: A smiling and a non-smiling image from the GENKI-4k dataset.

The next section describes Gaussian Derivatives briefly.

3 Gaussian Derivatives

Gaussian derivatives can efficiently describe the neighborhood appearance of an image for recognition and matching. This can be done by calculating several orders of Gaussian derivatives normalized in scale and orientation at every pixel.
The basic Gaussian function is defined as:

$$G(x, y; \sigma) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

Here σ is the scale factor or variance and defines the spatial support. This function measures the intensity of the neighborhood and does not contribute to the identification of the neighborhood and can be omitted. The first order derivatives are of the form:

$$G_x(x, y; \sigma) = \frac{\partial G(x, y; \sigma)}{\partial x} = -\frac{x}{\sigma^2} G(x, y; \sigma) \quad (2)$$

$$G_y(x, y; \sigma) = \frac{\partial G(x, y; \sigma)}{\partial y} = -\frac{y}{\sigma^2} G(x, y; \sigma) \quad (3)$$

First order derivatives give information about the gradient (intensity and direction). The second order derivatives are given by:

$$G_{xx}(x, y; \sigma) = \frac{\partial^2 G(x, y; \sigma)}{\partial x^2} = (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}) G(x, y; \sigma) \quad (4)$$

$$G_{yy}(x, y; \sigma) = \frac{\partial^2 G(x, y; \sigma)}{\partial y^2} = (\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}) G(x, y; \sigma) \quad (5)$$

$$G_{xy}(x, y; \sigma) = \frac{\partial^2 G(x, y; \sigma)}{\partial x \partial y} = \frac{xy}{\sigma^4} G(x, y; \sigma) \quad (6)$$

Second order derivatives provide us with information about image features such as bars, blobs and corners. Higher order derivatives are only useful if the second order derivatives are strong otherwise they just contain image noise.
Normalizing Gaussian derivatives in scale is not a trivial task. Several methods have come up in the past addressing this problem. It was suggested by Lindeberg in [14] that Gaussian derivatives be calculated across scales to get scale invariant features and then Lowe in [15] defined the intrinsic or characteristic scale as the value of the scale parameter at which the Laplacian provides a local maximum. The computational cost of directly searching the scale axis for this characteristic scale can be prohibitively expensive. A cost-effective method for computing multi-scale Gaussian derivatives is described in the following section. The inverse-tangent of the ratio of first order derivatives at any image point is considered to be the direction of the gradient. It has been shown that Gaussian derivatives are steerable [16] and by using appropriate trigonometric ratios the Gaussian derivatives can be rotated in the desired direction.

4 Half-Octave Gaussian Pyramid (Multi-scale Gaussian Derivatives)

This algorithm has been discussed in detail in [17] and an integer coefficient version of the same can constructed using repeated convolutions of the binomial kernel (1, 2, 1).
The algorithm involves repeated convolutions with a Gaussian kernel in a cascaded configuration where the process is speeded up by approximating a Gaussian filter with separable binomial filters as shown below:

$$G(x, y; \sqrt{2}) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad (7)$$

A key feature of this algorithm is that for different levels of the pyramid the difference of adjacent image pixels in the row and column directions are equivalent to convolution with Gaussian derivatives.
The pyramid is very easy to access, derivative values can be determined for every image position by using bilinear interpolation and derivatives between scale values can be computed using quadratic interpolation between adjacent levels of the pyramid. The following sets of equations explain how different order of derivatives can be calculated using difference of adjacent image pixels in the row and column directions:

$$\frac{\partial p(x, y, k)}{\partial x} = p * G_x(x, y; 2^k \sigma_0) \approx p(x+1, y, k) - p(x-1, y, k) \quad (8)$$

ISBN: 978-1-61804-164-7

$$\frac{\partial p(x,y,k)}{\partial y} = p{*}G_y(x,y;2^k\sigma_0) \approx p(x,y{+}1,k){-}p(x,y{-}1,k) \tag{9}$$

$$\begin{aligned}\frac{\partial^2 p(x,y,k)}{\partial x^2} &= p * G_{xx}(x,y;2^k\sigma_0) \\ &\approx p(x+1,y,k) - 2p(x,y,k) + p(x-1,y,k)\end{aligned} \tag{10}$$

$$\begin{aligned}\frac{\partial^2 p(x,y,k)}{\partial y^2} &= p * G_{yy}(x,y;2^k\sigma_0) \\ &\approx p(x,y+1,k) - 2p(x,y,k) + p(x,y-1,k)\end{aligned} \tag{11}$$

$$\begin{aligned}\frac{\partial^2 p(x,y,k)}{\partial x \partial y} &= p * G_{xy}(x,y;2^k\sigma_0) \\ &\approx p(x+1,y+1,k) - p(x+1,y-1,k) \\ &\quad - p(x-1,y+1,k) + p(x-1,y-1,k)\end{aligned} \tag{12}$$

In the above equations at the k_{th} level of the pyramid the support is defined by $\sigma_k = 2^k\sigma_0$ and the image at the same level is defined by $p(x,y,k)$.
The next section is about dimensionality reduction using Principal Component Analysis(PCA) and why we need it.

5 Principal Component Analysis

In our experiments the part of the image containing the face was normalized to 64 X 64 pixels, this size of 64 X 64 pixels for the normalized region was chosen after extensive experimentation where normalized images of 64 X 64 pixels gave better results at smile detection as compared to other sizes. We want to calculate several orders of derivatives at 2 levels of scale for every pixel but that would lead to very large feature vectors which will be difficult to handle. So we divide the image into cells of 4 X 4 pixels and the feature vector contains the mean and standard deviation of the descriptor values (gaussian derivatives) for each cell of 4 X 4 pixels.

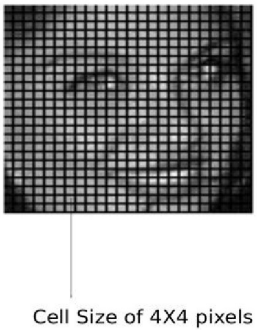

Figure 2: The image divided into cells of 4 X 4 pixels.

Principal component analysis was then used to get rid of correlated dimensions by transforming the original dimensions into new dimensions which are a linear sum of the original dimensions but are linearly uncorrelated and then these new dimensions are ranked according to the variance i.e. the dimension which accounts for the most variability in the data gets the first rank and so on [18].

PCA is done by eigenvalue decomposition of the data correlation matrix after normalizing the data for each dimension. PCA provides you with scores and loadings. The scores are the transformed values corresponding to your data point and loadings the coefficient your original variable should be multiplied with to get the score.

We found that just 61 out of the 3920 dimensions can account for most of the variability in data and give us an acceptable accuracy. Support Vector Machines were then used to discriminate between different poses. In section seven we compare the execution time of the SVM with and without using PCA.

6 Support Vector Machines

Support Vector Machines (SVM) belong to a family of generalized linear classifiers and can be interpreted as an extension of the perceptron [9].
After using several types of kernels we settled on the radial basis kernel as it provided us with the maximum accuracy, represented by the following equation:

$$K(x_i,x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{13}$$

The SVM employed was a soft margin SVM, soft margin SVMs are used when the classes are not separable even after transforming the data to a higher dimension. The condition for the optimal hyper-plane can be relaxed by including an extra term ξ [19]:

$$y_i(X_i^T W + b) \geq 1 - \xi_i, (i = 1, ..., m) \tag{14}$$

ISBN: 978-1-61804-164-7

For minimum error,ξ_i should be minimized as well as $||W||$, and the objective function becomes:

$$minimize \quad W^T W + C\sum_{i=1}^{m}\xi_i^k$$
$$subject\,to\, y_i(X_i^T W + b) \geq 1 - \xi_i,\ and\ \xi_i \geq 0; (i = 1, ..., m) \tag{15}$$

Here C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error. And σ is the width of the radial basis kernel. The C-penalty parameter was chosen using cross validation. For the data in hand the value C=7 and $\sigma = 81$ lead to the highest classification accuracy.

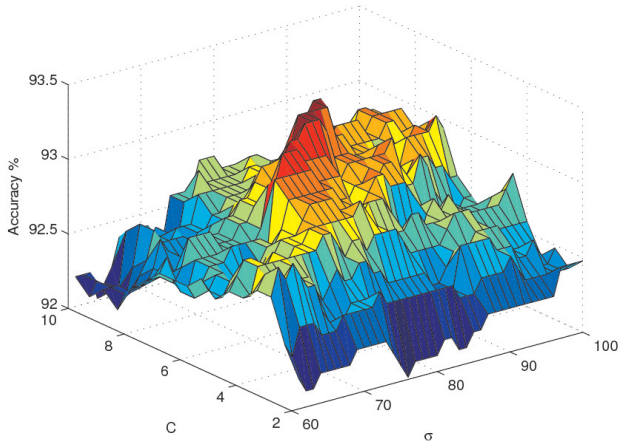

Figure 3: Graph of Classification Accuracy vs. C-parameter and σ.

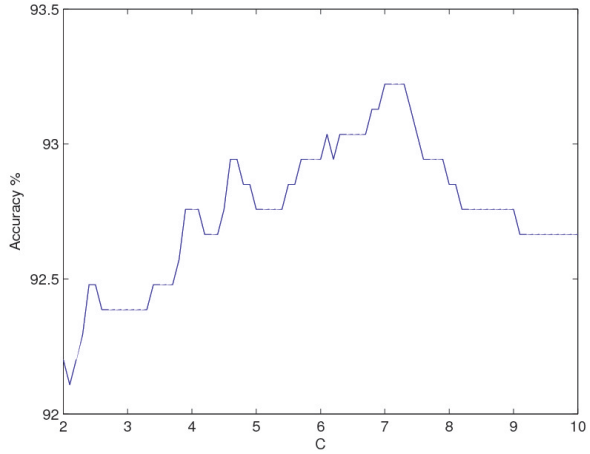

Figure 4: Graph of Classification Accuracy vs. C-parameter at $\sigma = 81$.

SVM can also be used for probabilistic decision making by using the distance of an instance from the separating hyperplane to measure the posterior probalility $p(j|x)$ where x is the distance of the instance from the hyperplane and j the class to which the data may belong to. This concept has been used in section seven to measure smile intensity.

7 Experimental Procedure and Results

We used 3577 out of the 4000 images in the GENKI-4k dataset removing ambiguous cases and images with serious illumination problems like partial lighting of the face.
Face detection was then performed on the images in the dataset using the OpenCV face detector [20]. Following that a half-octave gaussian pyramid was constructed over a normalized imagette of the face which is of the size 64X64 pixels.
Sixty percent of the data for training and the rest for testing.The data was split several times and the accuracy calculated for every split and finally the average was calculated. We achieved a classification accuracy of 92.97% using PCA for dimensionality reduction. Our results are superior to the state of the art Gabor Energy Filters as shown in the table below.

	GEF	MGD with PCA
Accuracy%	90.78	92.97

Table 1: Our accuracy compared with the accuracy obtained using Gabor Energy Filters(GEF)

Figure 5 shows the ROC for the SVM trained on the GENKI-4k dataset.

ISBN: 978-1-61804-164-7

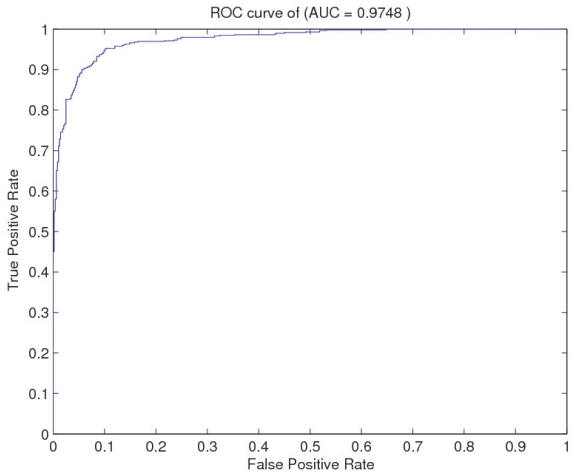

Figure 5: ROC curve for our smile detector.

Table 2 shows the prediction times with and with using PCA, as we can observe the use of PCA speeds up the prediction time by a factor of over 60.

	SVM with PCA	SVM without PCA
Prediction time(sec)	0.254	17.133

Table 2: Comparison of prediction time with and without using PCA

Then we use our SVM trained over the GENKI-4k datset and use it over the Cohn-Kanade dataset for the purpose of validation. We use sequences of people smiling and the SVM was used to give a probabilistic output for each frame of the sequence, it shows how these probaility estimates can be used to measure the intensity of smiles.

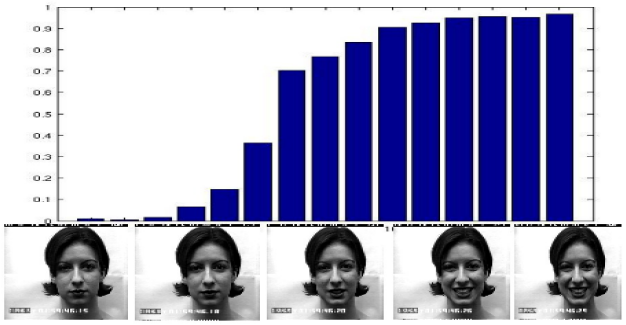

Figure 6: Smile intensity using probability estimates.

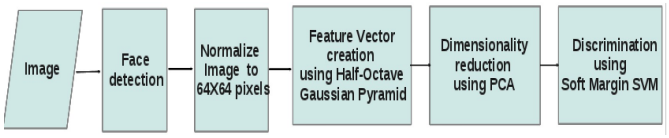

Figure 7: Schematic of our approach.

8 Conclusion

We have shown that our approach using Multi-scale Gaussian Derivatives performs better than Gabor Energy Filters for Smile detection. This approach can be extended to other facial expressions as well.
We have also shown how PCA can be used for dimensionality reduction which then leads to a huge reduction in the prediction time.
Finally, we have shown how probabilistic estimates produced by a Support Vector Machine can be used to estimate smile intensity.
Codes exist for calculating Multi-scale Gaussian Derivatives effectively on embedded systems hence our approach can be easily ported to handheld devices like cameras and cellphones.

References:

[1] T. Kanade Y. Tian and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[2] Y. Qi A. Kapoor and R. Picard. Fully automatic upper facial action recognition. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.

[3] I. Kotsia and I. Pitas. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *IEEE Transactions on Image Processing*, 16(1):172–187, 2007.

[4] Z. Wen and T. Huang. Capturing subtle facial motions in 3d face tracking. In *Proceedings IEEE International Conference on Computer Vision*, 2003.

[5] B. Wu C. Huang Y. Wang, H. Ai. Real time facial expression recognition with adaboost. In *Proceedings 17th International Conference on Pattern Recognition*, 2004.

[6] M. Frank C. Lainscsek I. Fasel M. Bartlett, G. Littlewort and J. Movellan. Fully automatic

ISBN: 978-1-61804-164-7

facial action recognition in spontaneous behavior. In *Proceedings Automatic Facial and Gesture Recognition*, 2006.

[7] I. Fasel J. Susskind G. Littlewort, M. Bartlett and J. Movellan. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615–625, 2006.

[8] Marian S. Bartlett T. Wu and Javier R. Movellan. Facial expression recognition using gabor motion energy filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 42–47, 2010.

[9] V.N. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.

[10] J. Cohn T. Kanade and Y.-L. Tian. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 46–53, March 2000.

[11] P. Ekman and W. Friesen. Pictures of facial effect. Technical report, Photographs available from Human Interaction Laboratory, University of California, San Francisco, 1976.

[12] http://mplab.ucsd.edu. The MPlab GENKI database, GENKI-4K subset.

[13] I. Fasel M. Bartlett J. Whitehall, G. Littlewort and J. Movellan. Towards practical smile detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2106–2111, 2009.

[14] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Press, 1994.

[15] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999.

[16] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[17] J.L. Crowley and O. Riff. *Springer Lecture Notes on Computer Science*, volume 2695, chapter Fast Computation of Scale Normalized Gaussian Receptive Fields, pages 584–598. 2003.

[18] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.

[19] C. Cortes and V.N. Vapnik. *Machine Learning*, volume 20, chapter Support-Vector Networks, pages 273–297. Springer, 1995.

[20] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 20(17):137–154, 2004.